Constructing a ROCKS Cluster

Ryan Wojtyla

November 12, 2018

1 Preparation

Preparation is key before attempting to build a ROCKS cluster; there are many steps to take care of beforehand.

1.1 Backup Data

One of the most important preliminary steps is to first backup all the data on the devices that are being updated. I backed up the contents of the CE, minus `/mnt` and `/cvmfs`, the SE, and both the server and data partitions of NAS-0 onto NAS-1. For the CE and SE, `/mnt` was excluded because I don't want to backup both of the NASs' data onto NAS-1. `/cvmfs` was excluded for the CE because it is externally mounted from CERN, thus eliminating the need for a local backup. Unlike the CE and SE, NAS-0 does not have NAS-1 mounted, so the `rsync` command had to remotely copy the NAS-0 files from the CE. Since there is already an older backup of NAS-0 on NAS-1 and there is not enough space on NAS-1 to fully copy another backup of NAS-0, all of the newer files on NAS-0 were manually copied into the old backup of NAS-0, effectively updating it.

- Backup the CE (command run while logged into the CE):
  - `rsync -av --exclude '/mnt*' --exclude '/cvmfs' / /mnt/nas1/CEBackup-20180`
- Backup the SE (command run while logged into the SE):
  - `rsync -av --exclude '/mnt' / /mnt/nas1/SEBackup-20180803`

- Backup the server partition of NAS-0 (command run while logged into the CE):
  - `rsync -av --exclude '/nas0' root@10.1.255.234:/ /mnt/nas1/nas0-bak-201807`

Ideally, the entirety of one compute node would also be backed up at this point, but due to catastrophic hardware failures preventing the nodes from being run (one of the big Tripplite UPSs AND the APC UPS have failed, leaving me with only one UPS), I'm going to come back to them later.

1.2 Create Boot Drives

Once all of the data is secure, the ISO images of the ROCKS rolls can be downloaded and installed. The necessary ROCKS rolls, available here, are:

- kernel
- base
- core
- CentOS
- Updates-CentOS
- ganglia
- htcondor

These ISO images may be directly placed on a formated USB using the `dd` command. The computer-given name of the USB drive of the form `/dev/sd?` (`/dev/disk?` on MacOS) must be determined. On Linux, the `lsblk` command may be used to discover this proper name. On Mac, the "Disk Utility" application may be used.

To create the bootable USB, the `dd` command may be used. The USB must be first unmounted, `sudo umount /dev/sd?`, then the following command may be run: `sudo dd bs=4M if=INPUT-FILE.iso of=/dev/sd? conv=fdatasync` (for MacOS: `sudo dd bs=4m if=INPUT-FILE.iso of=/dev/disk? && sync`). The `bs` option determines the block size of each packet transferred at a time, the input file is the ISO file of the image to be copied,

and the output file is the name of the USB to be made bootable. Both the `conv=fdatasync` and `&& sync` parts of the commands ensure that all data has been written before the next packet is sent.

Since coming across eight USBs may be a challenge, the USBs will be created as they are needed. The first USB to be made is the one containing the "kernel" roll.

Actually, the firmware doesn't support booting from USB, so I need to go get DVDs because CDs are too small.

All the rolls were burnt onto separate DVDs except for the large "CentOS" roll, which is more than 7GB, larger than the 4.7GB discs.

1.3 Ensure NAS-1 Remains Operational

Because NAS-1 does not need to be updated, and since everyone needs to access their data on it, it must be kept running and accessible during the rebuild process. Since NAS-1 has never technically been part of the cluster with ROCKS, it is independently addressable via its own IP address. This independent addressability permits NAS-1 to be mounted and ssh-ed into without the rest of the cluster. It just needs to be left on and connected to the router.

2 Installation

2.1 Kernel Installation

The instructions say to first use the "kernel" disc. The CE was turned on so that the optical drive has power to open, the disc was quickly inserted, and the CE was allowed to continue booting. It automatically booted into the disc, and it applies checks before continuing.

Once it has finished booting, the system language must be selected. After that, regional information, such as timezone, is entered.

2.2 Network Configuration

Select *Network & Host Name*.

Now, the network must be configured. The option for the 10GB/s ethernet connection is turned on.

Next, the "Configure..." button is selected. Under the "IPv4 Settings" tab, the "Method" is set to *Manual*, and the IP of the CE, 163.118.42.1, is entered. The netmask, an arbitrary range of IPs for the network, is set to 25 in order to encompass the IP range 163.118.42.1-163.118.42.127. The gateway is the IP address of the router through which the computer is connected to the network. It was found by `ssh`-ing into NAS-1 and running `ip route`, which displays the default gateway IP, 163.118.42.126, and the IP of NAS-1, itself. Under the "IPv6 Settings" tab, the "Method" is set to *Link-Local Only*.

With this information entered, the *Save* button is pressed. The ethernet connection may be switched on as before, and the "Current host name" in the bottom right of the screen should read "uscms1.fltech-grid3.fit.edu", the cluster's hostname. If it reads something else, enter the correct hostname on the bottom left of the screen. Once everything here is completed, *Done* may be selected.

With network configuration out of the way, all the previously grayed-out options are now full of color! Now the local network needs to be configured. Select *Cluster Private Network*.

The "Private Cluster Interface" dropdown menu is a list of the ethernet ports, similar to the list for selecting the public network connection. Since the 10GB/s connection is used for the public connection, the normal ethernet port is used for the private connection. Refer back to the "Network & Hostname" section to see which port is occupied, and select that one in the "Private Cluster Interface" dropdown menu. All the other fields should not need to be altered; the default settings are fine and expected.

2.3 Rolls

Now the rolls can be installed! Select "ROCKS ROLLS". Here, rolls may be installed remotely from the ROCKS website. Select *List Available Rolls* to generate the roll list. Select the following rolls:

- base
- CentOS
- core
- ganglia

- htcondor
- kernel
- perl
- python
- Updates-CentOS-¡version¿

Once all the desired rolls are selected, select *Add Selected Rolls*. After the rolls appear in the "Selected Rolls" section, select *Done*.

If no network rolls can be found, change "Network Roll" to "CD/DVD" Roll, and just install the "kernel" roll. The other rolls may be installed later.

2.4 Cluster Configuration

Select *CLUSTER CONFIG* to configure the cluster. The fields are filled as follows:

Cluster Name	USCMS-FIT-Grid
Contact	¡administrator email¿
Project URL	`https://research.fit.edu/hep/hohlmann-research-group/grid-`
Latitude/Longitude	N28.0622 W80.6237
Certificate Organization	OSG
Certificate Locality	Melbourne
Certificate State	Florida
Certificate Country	US
NTP Servers	pool.ntp.org

After the fields have been verified to be correct, select *Done*.

2.5 Partitioning

Now the CE must be partitioned. Select *INSTALLATION DESTINATION*. Select both drives from the menu, and select *I will configure partitioning.* under "Other Storage Options". Select *Done* to be kicked over to Anaconda's partitioning system.

First, the old partition, the CentOS 6 one, was selected, then removed by selecting the "-" at the bottom of the window. Then the *Click here to create them automatically* option was selected.

The created `/home` partition was removed, because NAS-0 is what holds our home directories, and it was renamed `/export` and given 100GB. The default `/boot` and `swap` partitions were left untouched, the `/var` partition was created and given 100GB, and the `/` partition was allocated the remainder of the available space, 256.76GB. The curious "Unknown" section of the other 233GB drive was not touched.

Once everything is together, select *Done* and carefully review the changes. If the changes are correct, select *Accept Changes*.

2.6 Commit

If everything is undoubtedly confirmed (to the best of your ability), and you're absolutely ready to destroy everything and build it back up, say a quick prayer to your deity of choice and select *Begin Installation*. WARNING: there is no "confirm" to *Begin Installation*

Create a root password when prompted, and do NOT setup a user.

3 Configuration

4 Troubleshooting

4.1 Boot Loader Install Failed

During installation, the boot loader failed to install. Selecting *no* on the dialogue box will present an "unknown error" screen with a log and a debug option. Selecting "debug" kicks the screen over to the terminal screen.

Navigating to the "program-log" tab, the following errors may be found after the installer attempted to run `grub2-install --no-floppy /dev/sda`:

- `grub2-install: warning: Attempting to install GRUB to a disk with multiple partition labels. This is not supported yet...`
- `grub2-install: error: embedding is not possible, but this is required for cross-disk install.`

The decision to try to install to `/dev/sda` is strange because according to `fdisk -l /dev/sda` and `fdisk -l /dev/sdb`, run in the "shell" tab, reveal that `/dev/sdb1` is marked for boot, whereas no such marking is present in the single part of `/dev/sda`.

Trying to run `grub2-install --no-floppy /dev/sda` and `grub2-install --no-floppy /dev/sdb` produces the same result: `grub2-install: error: /usr/lib/grub/i386-pc/modinfo.sh doesn't exist. Please specify --target or --directory.` Investigating that path, `/usr/lib/grub` comes up empty. To find where that file is actually located, since it has to be somewhere if the installer is running that command, I ran `find / -name modinfo.sh`. It appears in two places: `/mnt/sysimage/boot/grub2/i386-pc/modinfo.sh` and `/mnt/sysimage/usr/lib/grub/i386-pc/modinfo.sh`.

I'm trying to run the installer's command again while specifying the location of the file it needs: `grub2-install --no-floppy --directory='/mnt/sysimage/boot/grub2 /dev/sda`. Now it says: `grub2-install: error: cannot open '/mnt/sysimage/boot/grub2/ No such file or directory.`. Hmm, let's see if that file exists anywhere with: `find / -name kernel.img`. Ahh, it's at `/mnt/sysimage/usr/lib/grub/i386-pc/kernel.i` Let's run the installer's command again, but the directory will be changed to `/mnt/sysimage/usr/lib/grub/i386-pc`: `grub2-install --no-floppy --directory='/mnt/sysimage/usr/lib/grub/i386-pc' /dev/sda`. Ah ha! Now I'm getting the same error as the installer! Let's try with `/dev/sdb`: `grub2-install --no-floppy --directory='/mnt/sysimage/usr/lib/grub/i386-pc /dev/sdb`. Ohh! `Installation finished. No error reported.`!

Since the installer crashed, the CE needs to be rebooted. Be sure to swiftly eject the disc when it first begins to power on so that it tries to boot normally. It will boot to grub rescue mode because, while grub is installed, it is not configured correctly.

Running the `ls` command in grub rescue will show that there are two drives, (hd0) and (hd1), and `set` will reveal that it is trying to boot into `(hd0)`. Running `ls (hd0)` will show that its file system is unknown, while `ls (hd1)` shows a recognizable one, such as ext2. The file system of (hd0) is unknown because it is the hot spare for the RAID; grub is trying to boot into the hot spare, just as it was trying to do during installation.

The `set` command said that the variable `prefix` was set to `(hd0)/boot/grub2` and `root` was set to `hd0`. To point those at the right place I ran the following two commands:

- `set prefix=(hd1)/boot/grub2`

- `set root=hd1`

Instructions online say after fixing the variables to try loading the normal module with `insmod normal`, but I get the error, `error: file '/boot/grub2/i386-pc/normal.m not found.`. Running `ls (hd1)/` shows me nothing! Does that mean there's nothing on the drive I need, or just that the information isn't quite accessible? Hmm. I guess a restart couldn't hurt more; let's try that now that the variables are set.

Actually, you know what, I think I discovered the mistake that caused this whole mess. I decided to check to box to include the hot spare in the installation process. It has no place here; it's managed strictly by the RAID card. It got assigned `sda` and messed everything up. I'm just gonna turn all this off and redo the installation correctly this time.

4.2 Remote ROCKS Servers Inaccessible

Normally, the ROCKS rolls are able to be accessed from the remote ROCKS server in the "ROCKS ROLLS" section of the installation. After the kerfuffle with the boot loader, however, the server seems to be inaccessible even though the CE has internet, verified by pinging through the console.

I'm going to use the ability of the installer to remotely obtain the rolls to see if the issue is just with the communicating with the server, or if it's a larger problem. I've placed all the required rolls on NAS-1, and I'm going to try accessing them from there.

To do that, the rolls need to be hosted on an HTTP server on NAS-1. I'm installing and setting up Apache.

When I tried to `yum install httpd`, none of the mirrors worked, so the installation failed. I tried a `yum update` and a `yum clean all`, but the clean just confirmed that there was a serious problem. Now yum reports that `mirrors.centos.org` cannot be resolved. `yum repolist all` also runs into the same issue; only the 5 recommended steps are listed.

Turns out `/etc/resolv.conf` was blank, and, amazingly, adding `nameserver 9.9.9.9` to it fixed the problem. I wonder if that same file is messed up in the installer.

On the installer, by changing the kernel parameters from "quiet" to "verbose", or even "debug", the booting up of the GUI installer can be seen and interrupted, providing access to the shell. `ifconfig` shows the four network ports and the local network, just as it should, but nothing

is connected to the internet. The cables are plugged in, so let's see about enabling the internet.

The Arch Linux wiki page for network configuration is very detailed and it seems to be helpful. I'm going through it.

I've gone through the page, setting the ip manually and editing `/etc/hosts`, but no luck. I'm going to try the `/etc/resolv.conf` thing that worked on NAS-1.

The ROCKS installation CD on the CE suddenly has internet! The IP is 163.118.42.1/25, the broadcast IP is 163.118.42.127, and the gateway IP is 163.118.42.126 (`ip address add 163.118.42.1/25 broadcast + dev enp10s0f0`). A route was added: `ip route add default via 163.118.42.126 dev enp10s0f0`. A nameserver was added to `/etc/resolv.conf`, `nameserver 9.9.9.9`. The installer can be restarted with `anaconda`.

Now the manual installation via anaconda's CLI can begin.

The locations of several important lines of code:

ROCKS Rolls GUI	`/usr/share/anaconda/addo`
GUI Function	`/usr/share/anaconda/addo`
Actual Function	`/opt/rocks/lib/`

It is difficult to test the individual lines and functions because of so many references to "self" in the code, which indicates an incredible reliance on being in a special running environment to operate properly.

I'm trying something new. I'm installing ROCKS onto a flash drive on another machine, then I'm going to `dd` the flash drive image onto the hard drive in the cluster. Maybe that will work.

Make sure the USB is COMPLETELY blank (`parted /dev/sd<x> mklabel loop`).

To get to the anaconda command line application:

- load the Anaconda GUI like normal
- Ctl-Alt-F2 to drop to the base terminal of the CD
- try to run `anaconda`
- when X fails to load properly, select the VNC option
- Although Anaconda will continue to not start up properly, the Anaconda environment (with the tmux bar at the bottom) will load.

Just to try something new, I loaded a ubuntu liveCD onto the CE and had a poke around. It looks like everything is where it used to be on the partitions of the original drive, `/dev/sdb2`, specifically the `/root` partition. A curious note about how those partitions are organized: `/dev/sdb2` is formated as a Linux logical volume, and the logical partitions of that logical volume are found under `/dev/rocks_uscms1/`. `/dev/rocks_uscms1/root` is the original root file system, and I mounted it with `mount /dev/rocks_uscms1/root /mnt/sdb_root`. Something interesting I found, though, is the contents of `/mnt/sdb_root/etc/redhat-release`, which report that CentOS 7 is installed on the system. Hmm. Did the original install do something? Additionally, the `/mnt/sdb_root/boot` directory is blank. While THAT boot directory is blank, however, the specific boot partition of `/dev/sdb`, `/dev/sdb1`, is fully populated with an `el7` image and an updated grub.

To enable ssh on a Ubuntu LiveCD, install `openssh-server` with `sudo apt install openssh-server`, and check to make sure the service is running with `sudo service ssh status`.

BREAK-THROUGH! After fiddling with GRUB from the LiveCD, namely reinstalling it onto the boot partition, the CE still refused to boot! This is because the 3Ware card was not configured to be booted into on the boot order section of the BIOS. This page is accessed by slamming the ¡Delete¿ key immediately upon turning on the CE. The 3Ware card had its boot status elevated, and I was presented with a fully fleshed out GRUB prompt! (In contrast to the extremely limited rescue GRUB prompt from earlier.) What is available is summarized in Table 4.2.

Figure 1: A list of the partitions available to us at the GRUB prompt.

Partition	Filesystem	Summary of Contents
`(proc)`	`procfs`	none
`(hd0)`	n/a	n/a
`(hd0,msdos2)`	n/a	n/a
`(hd0,msdos1)`	`ext*`	boot partition of CE data drive
`(hd1)`	`ext*`	none
`(hd1,msdos1)`	`ext*`	misc. config files (namely Squid)
`(fd0)`	n/a	n/a

Let's try to boot into the boot partition. Following some instructions

from online, the first step is to load the `ext2` module, `insmod ext2`. Next, the root of the prompt needs to be set to the boot partition, `set root='(hd0,msdos1)'`. If the root was set correctly, the output of the `ls /` command ought to be identical to the output of `ls (hd0,msdos1)`. In the new `/`, there should be two files that start with `vmlinuz`. We're only concerned with the one that does not have the word "rescue" in the name. To load the compiled kernel, `linux <non-rescuse vmlinuz>`, then to boot it, `boot`.

Unfortunately, the kernel panics when it tries to boot; the rescue kernel was also tried to no avail. The rescue kernel has the panic `not syncing: VFS: Unable to mount root fs on unknown-block(0,0)`. A quick search has revealed that it might be complaining about not having the `initramfs` loaded beforehand. We have that file, but how do we load it properly?

Running the setup commands in a different order, while throwing in a couple new ones, produces different results, but nothing successful.

`set pager=1`	
`set root=(hd0,msdos1)`	
`insmod ext2`	Load the module for dealing with the boot partition's filesy
`linux /vmlinuz root=/dev/sda1`	Load the desired `vmlinuz` image, usually not the "rescue" c
`initrd /initrd-plymouth.img`	Load the corresponding `initrd` image.
`boot`	Attempt to boot into the selected image.

Figure 2: An attempt to boot into the OS from GRUB.

Unfortunately, the `pager` setting does not page the output from `boot`, so we still have no clue what's going on after the button's hit.